Neurocomputing 253 (2017) 193–200

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multimodality semantic segmentation based on polarization and color images

Fan Wang*, Samia Ainouz, Chunfeng Lian, Abdelaziz Bensrhair

LITIS Laboratory, INSA de Rouen, Avenue de l'université, 76801 Saint-Etienne du Rouvray, France

ARTICLE INFO

Article history:
Received 31 May 2016
Revised 25 August 2016
Accepted 16 October 2016
Available online 8 March 2017

Keywords:
Semantic segmentation
Polarization
Joint boosting

ABSTRACT

Semantic segmentation gives a meaningful class label to every pixel in an image. It enables intelligent devices to understand the scene and has received sufficient attention during recent years. Traditional imaging systems always apply their methods on RGB, RGB-D or even RGB combined with geometric information. However, for outdoor applications, strong reflection or poor illumination appears to reduce the visualization of the real shape or texture of the objects, thus limiting the performance of semantic segmentation algorithms. To tackle this problem, this paper adopts polarization imaging as it can provide complementary information by describing some imperceptible light properties, which varies from different materials. For acceleration, SLIC superpixel segmentation is used to speed up the system. HOG and LBP features are extracted from both color and polarization images. After quantization using visual codebooks, Joint Boosting classifier is trained to label each pixel based on the quantized features. The proposed method was evaluated both on Day-set and Dusk-set. The experimental results show that using polarization setup can provide complementary information to improve the semantic segmentation accuracy. Especially, a large improvement on Dusk-set shows its capacity for intelligent vehicle applications under dark illumination condition.

1. Introduction

Semantic segmentation, which is also known as scene/image parsing or image understanding, aims to divide an image into predefined meaningful non-overlapped regions (e.g. car, grass, road, etc). As an important task in intelligent vehicle (IV) applications, its ultimate goal is to equip IV with the ability to understand the surrounding environment. Other IV tasks, such as pedestrian detection, obstacle detection or road surface estimation, could benefit from semantic segmentation.

The substantial development of image classification, object detection, and superpixel segmentation in the past few years have boosted the research in the supervised scene parsing. However, the challenges ranging from feature representation to model design and optimization are still not fully resolved. Up to feature extraction, most methods extract features from RGB or gray level images. Since local low-level features are sensitive to perspective variations, researchers tried to solve this problem through the multimodality manner, by combining some other information with RGB images to give a better performance, such as RGB-D images [1], and geometry information [2] etc. In another aspect, some special illumination cases, such as reflective surfaces (too bright) or dark shaded surfaces, would appear to cover real texture or feature information, hence limiting the algorithm's performance. Considering this limitation, we adopt polarization image as a new source of information, as multimodality image parsing algorithm, to improve the classification result.

Light is polarized once it is reflected from a surface. The light polarization properties are related to different surface materials, surface geometry structures, the roughness of the surfaces etc. So that these characteristics are coded implicitly in the light polarization state. In this point of view, polarization attributes can provide description of some surface features that can not be offered by color images. It is worth to know that, these attributes are still kept distinguishable under high reflection or in shadow areas, where the color-image based methods fail to produce reliable results.

In computer vision, there are many indoor polarization applications under ideal lighting conditions since early 1990s, e.g., surface modeling, shape recovery, and reflectance analysis. However, not much outdoor applications have been realized. The reason is that the outdoor incident and reflect light are extremely complex. To the best of our knowledge, no work in the literature has applied polarization in semantic segmentation, this is the first work which

* Corresponding author.
E-mail addresses: fan.wang@insa-rouen.fr, fan.vanee@gmail.com (F. Wang).

http://dx.doi.org/10.1016/j.neucom.2016.10.090
0925-2312/

attempts to utilize polarization information as features for outdoor image processing applications.

In this paper, we propose to combine the polarization images (resulted from polarization state of each pixel) with the color images to improve the accuracy of image semantic segmentation. The combination method, more specifically, is through the HOG, LBP and LAB features that are extracted on both the polarization images and the color images independently. These features are concatenated and feed into a joint boosting classifier, a feature selection based classifier known for its facility to integrate new sources of features. In the training process, the classifier randomly selects different polarization features and color features from the input space to produce the polarization-based semantic segmentation results. In comparison, we repeat the same algorithm, which extracts the HOG, LBP and LAB features on, however, only color images. After training another joint boosting classifier, the color-based semantic segmentation results are given. The comparison shows that the accuracy of the semantic segmentation is improved thanks to the included polarization features.

2. Background

2.1. Semantic segmentation

As very classical methods in image parsing, bottom-up semantic segmentation methods usually pursue the following pipelines [3]: (1) Grouping nearby pixels to image patches according to the local homogeneity. For this step, there exists methods like K-means, mean shift, Simple Linear Iterative Clustering (SLIC) [4], normalized-cut [5] etc; (2) Extracting local features, e.g., HOG, LBP, texture or curvature, from each patch; (3) Feeding the extracted features and hand-labeled ground truth to a classification model to produce a compatible score for each training sample; (4) Feeding testing samples to the trained classifier. After all these steps, a brute semantic segmentation result is obtained on the test image. To give a more coherent result, state-of-the-art methods commonly perform a global optimization based on Markov Random Field (MRF) or Conditional Random Field (CRF), in which training result acts as the unary term, while the pairwise term is defined over four- or eight-connected neighborhood.

Numerous methods accomplish the semantic segmentation through these four steps. Superpixels are frequently used as they are more natural and efficient in representation than pixel level features, since the latter one is ambiguous and sensitive to noise. There are some recent efficient and good performing superpixel generation methods as TurboPixel [6], SLIC superpixels [4] etc. For local features, scale-invariant feature transform (SIFT) feature [7], histogram of gradient (HOG) feature [8], local binary patterns (LBP) [9] and textons [10] are widely used. Shotton et al. [10] proposed to use texture layout filters with textons to represent the local texture layout information. Tighe et al. [2] tried to combine global and local features to nonparametrically classify over retrieved similar exemplars. Recently, deep convolutional neural network learned features [11] have been applied to replace the hand craft features which achieved promising performance.

Torralba et al. [12] proposed an efficient Joint Boosting classifier based on sharing features between different classes. This classifier fits well the semantic segmentation problem, as it enables to classify large scale of different object classes, and to describe many different views of the object. Shotton et al. [10] applied the texture layout filter in this classifier, and proposed an optimized search instead of the original best search to accelerate the computation. Costea and Nedvschi [13] used multi-features (HOG, LBP and Color feature) to replace the texture features in the texture layout filter, and applied the original version of Joint Boosting as in [12] at over 50 FPS. In this paper, we still use the same type of features as that used in [13]. While, the difference is that the feature extraction procedure is applied in the superpixel level, instead of sampling pixels from image grid. Additionally, we include polarization features into classification in our algorithm. We employ the Joint Boosting classifier used in [10], along with an optimal strategy to reduce threshold searching space.

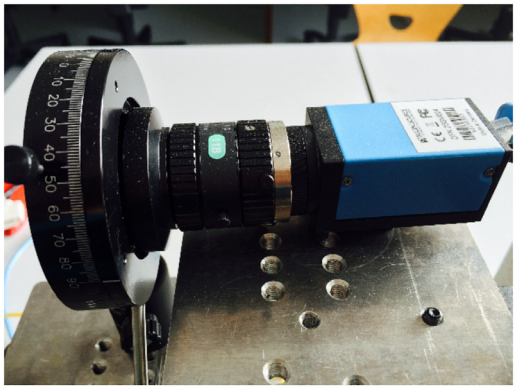

Fig. 1. A manually polarization imaging system. A degree labeled linear polarization filter is positioned in front of a blue CCD camera. Filter should be turned to three different degrees along with an image taken in each degree.

2.2. Polarization theory

Polarization-sensitive imaging systems emerged at early 1990s [14]. This imaging system quickly becomes very attractive for computer vision research since it reveals important information about the physical and geometrical properties of the targets [15]. Many imaging polarimeters have been designed in the past for several fields, ranging from metrology [16], image segmentation [15,17] to PolSAR application [18].

Common polarimetry imaging systems, which measure the polarization state of the out-coming light across a scene, are mainly based on a Polarization State Analyzer (PSA) in front of a CCD camera (see in Fig. 1). Real-time polarimetry camera for IV application exists as [19]. These systems permit to acquire the three dimensional structured Stokes vectors at pixel level, which fully describe the out-coming light properties[1].

Stokes vector S fully characterizes the time-averaged polarization properties of a radiation. It is defined by the following combination of complex-valued components E_x and E_y of the electric field, along two orthogonal direction x and y [20]:

$$S = \begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} \langle E_x, E_x^* \rangle + \langle E_y, E_y^* \rangle \\ \langle E_x, E_x^* \rangle - \langle E_y, E_y^* \rangle \\ 2Re\big(\langle E_x, E_y^* \rangle\big) \\ 2Im\big(\langle E_x, E_y^* \rangle\big) \end{pmatrix} \tag{1}$$

where S_0 represents to the total intensity of the optical field, S_1 is the tendency that the wave to look like horizontal (if positive) or vertical (if negative), S_2 similar but in 45^o if positive and 135^o if negative, S_3 reflects the nature and rotation direction of the wave. It is straight forward that (S_3 is ignored as stated in the footnote):

$$\begin{cases} S_0 > 0 \\ S_0^2 \geq S_1^2 + S_2^2 \end{cases} \tag{2}$$

The condition in (2) is the physical condition that Stokes formalism. An arbitrary vector that does not satisfy (2) is not a Stokes vector and does not possess any physical meaning.

[1] Stokes vector is originally 4-dimensional, while the last dimension describes the circular polarization part and it is almost near zeros, thus it is disregarded in many computer vision applications.

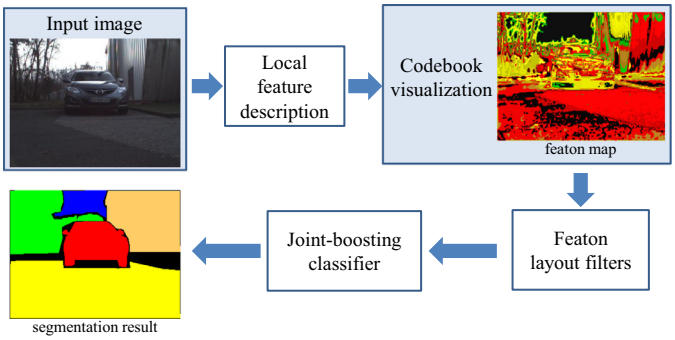

Fig. 2. Flowchart of the proposed method.

The wave reflected by target, noted as S_{in} is analyzed by the PSA at three independent state. In Computer vision application, a PSA is a linear polarize filter rotating at least three different angle $(\theta_i)_{i=1,2,3}$. The complete measurement can be written as:

$$I = AS_{in} \tag{3}$$

here I is the 3×1 matrix captured through the camera. It's not difficult to extract S_{in} given matrix A under ideal case [20]:

$$A(\theta_i) = \begin{bmatrix} 1 & -\cos^2 2\theta_i & -\frac{1}{2}\sin 4\theta_i \end{bmatrix} \tag{4}$$

Degree of polarization (DOP) and Angle of polarization (AOP) are other physical parameters which also characterize light properties. DOP stands for the percentage of polarized part among the overall intensity and AOP is the polarization angle compared with the reflection surface. These two parameters can be obtain through Stokes vector using:

$$\begin{cases} DOP = \sqrt{S_1^2 + S_2^2}/S_0 \\ AOP = tan^{-1}(S_2/S_1) \end{cases} \tag{5}$$

There are some early researches aim to identify material type and surface orientation by DOP and AOP [14,21], since degree of polarization is related to the material conductivity, while angle of polarization is related to the reflection surface. However, their goals at that time do not correspond to the supervised image parsing algorithms. Some segmentation methods were discussed in [15,17], they performed an image segmentation through polarization aided or based clustering without semantic interpretation. In the domain of PolSAR image understanding, they performed aerial image understanding by polarization, but they had more complementary information other than Stokes vector [18].

3. Polarization applied on semantic segmentation

In this section, we describe the proposed multimodality semantic segmentation algorithm using polarization and color images. This method follows four steps as shown in Fig. 2. First, we use local descriptors to describe the input image. This step is applied on both polarization and color images, so as to integrate information via different sources. These local descriptor vectors are then quantized through a clustered codebook which formulates the codebook maps as Fig. 3. As the final step, the pixel level classification is based on the TextonBoost [10] method, which contains two parts, being the texton-layout filters and the shared boosting classifier.

3.1. Local feature description

As the first step, we use local descriptors to describe the input image. Instead of the 17 dimensional texture features employed by Shotton et al. [10], we use the HOG, LBP and Lab color feature as proposed in [13]. The reason is that the combination of multiple descriptor types has shown improvement of classification accuracy [22]. HOG and LBP features are computed from gray scale images with Gaussian smooth of $\sigma = 0.25$, and also are computed from color features on RGB images with Gaussian smooth of $\sigma = 1.0$. The 24-dimensional HOG feature, 24-dimensional LBP feature and 3-dimensional color feature are extracted correspondingly. We denote *HLC feature* as the combination of HOG, LBP and Color features. The same configuration in [13] is applied here since these parameters are optimized to fit the real-time application.

As polarization images can potentially provide complementary information, in this step, both polar and color images are described. More specifically, The DOP and AOP are densely computed for every pixel, producing DOP image and AOP images as shown in Fig. 3. Regarding DOP and AOP as gray scale images, HOG and LBP features are extracted separately from them, which produces four polar-features: DOP-HOG, DOP-LBP, AOP-HOG and AOP-LBP. We take each polar-feature to combine with the HLC features separately, so as to make comparison and find the best polar-feature. As the computation of the DOP and AOP images involves at least 3 images, they are even more noisy than color images. A gaussian smoothing is thus applied, with $\sigma = 1.5$ which is larger than $\sigma = 1.0$ used for color features, before further computation.

3.2. Codebook visualization

The local descriptor vectors achieved through the previous step are then quantized through a clustered codebook. This is a common process employed by most of the modern semantic segmentation approaches, considering that the quantized vectors are known to be invariant to small changes.

More precisely, a large set of descriptor vectors are sampled and then trained using K-means clustering. The resulting centroids are recorded as visual words of the codebook. Once we have the codebook, a new feature vector can be matched to the most similar visual words of the codebook. The Euclidean distance is used to measure the similarity. By matching densely the local descriptors from one image, a codebook map is then generated. Fig. 3 illustrates the codebook maps of an image from HLC combined with AOP-LBP feature. This process reduces the high-dimensional feature vector to a single discrete value, thus making the post-processing be more efficient. The quantized feature vectors are invariant to small changes as similar descriptor vectors tend to be attached to the same visual word.

We use 50 visual words for the codebook as in [13], which is less than usual size in the literature that varies from 100 to 4000. Since that we do not have a large data set while we also have multi-feature descriptors, and outdoor scene are less complex both in terms of variety and textures. In every experiment, we have 3 features as HLC features plus one polar feature. we train 50 visual words for every feature, and get 200 visual words in total. After obtaining all the visual words, every pixel in the image is matched with the most similar visual word. Using this process, we get four images which are named featon maps (similar to the texton map in [10]), with each featon map corresponds to a feature.

3.3. Feature-layout-filters

After the quantization of the pixel level descriptor, we apply the texton-layout-filters, which is the first part of the TextonBoost algorithm [10]. As in this paper, it is applied on combined features instead of the texture features, we call it the feature-layout-filters to avoid ambiguity, but the formation of the filters is kept the same as that in [10]. For the same reason, the texton map (visualized descriptor) generated from the last step is renamed as featon map.

The main idea of this filtering is that a pixel is classified based on visual word counted in specific regions around the pixel. Each

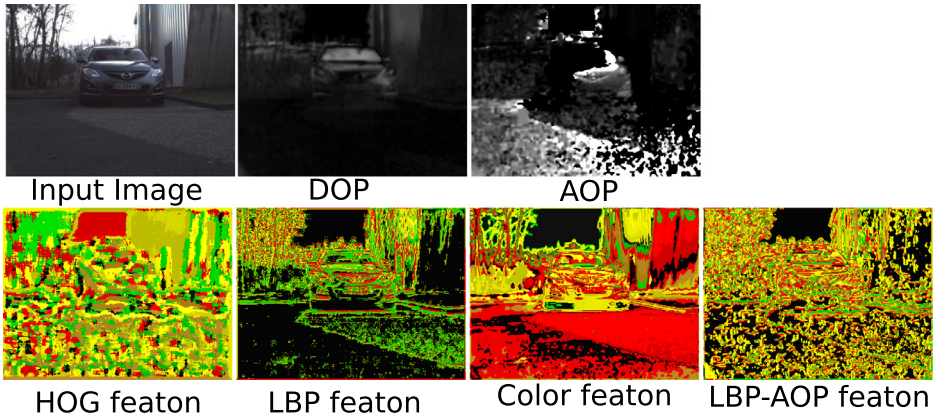

Fig. 3. Example of an input image with its DOP and AOP images, and the second row shows its featon image obtained via feature quantization.

feature-layout filter is a pair (r, f) of an image region r and a featon $f(=1,\ldots,50)$. A set of R which contains $N=50$ rectangles are randomly generated, such that their top-left and bottom-right corners lie inside a region of 200 × 200 pixels. These rectangles are defined using the relative coordinate of the pixel, which means that giving a rectangle r_n, $n \in N$ and pixel i, the position of the rectangle turns to be $r+i$. As recommended in [10], any configuration covering over half of the image size could be used. The feature response at any location i is the proportion of pixels under the offset region $r+i$ that have featon index f using:

$$v_{[r,f]}(i) = \frac{1}{area(r)} \sum_{j\in(r+i)} [T_j = f]. \tag{6}$$

The featon map is separated into $4 \times 50 = 200$ channels. For each channel, the integral image [23] T_i is used to efficiently compute in R over the whole image. This process will result in a 10,000 possible classification features.

3.4. Joint-boosting classifier

The joint-boosting method is used in the classification process, which is also the second part of the TextonBoost [10] algorithm. The algorithm iteratively selects discriminant featon-layout filters responses as 'weak learners', which is shared between a set of classes C. As each weak learner is shared between several classes, this classifier is known to be efficient for multi-class classification. Meanwhile, in this paper, the filter response is the concatenation given by the polar and color featon maps. As in each iteration, a discriminant filter response is compared and selected between polar and color ones, it also realizes the step of information fusion.

More specifically, in the mth training turn, for the cth class and on the most dominant feature dimension i, a weak learner $h_i^m(c)$ is learned:

$$h_i^m(c) = \begin{cases} a[v_{[r,t]}(i) > \theta] + b & if c \in C, \\ k^c & otherwise, \end{cases} \tag{7}$$

where $v_{[r,\,t](i)}$ is the feature computed from (6); while, a, b, θ, and k^c are the parameters that given through the training process.

After M turns, a 'strong' classifier is added up by the trained weak learners as $H(c,i) = \sum_{m=1}^{M} h_i^m(c)$. To reduce the computational cost, exhaustive feature search for each weak learner $h_i^m(c)$ is replaced by random feature selection. Thus, in each turn, the algorithm examines only a randomly chosen fraction $\zeta \ll 1$ of the possible features. It has been stated in [10] that the randomization not only speeds up learning, but also improves generalization by preventing over fitting to the training data.

4. Efficient application

In the real application, we apply two strategies regarding the time efficiency of the algorithm during the training process.

Firstly, we propose to apply a pixel sampling process before feeding all features into the training model. The reason is that using all the pixels in the image is too much consuming, and that neighboring pixels always carry similar information. In [10], a center pixel subsampling was performed over 3 × 3 or 5 × 5 grid to reduce training samples. Since this process is almost a random selection, it may sometimes select pixels which are noisy or less informative. In this paper, we propose to sample pixels using superpixel segmentation method, considering that a superpixel naturally represent a group of pixels which share similar features, which is fast and also robust. The SLIC superpixel [4] is used since it is a state-of-the-art superpixel segmentation method which is known as both fast and accurate. As a consequence, for one superpixel, each of 10,000 dimensions of the feature, a gravity centroid is computed among all the pixels inside. The final feature vector for a superpixel is the gravity centroid of the feature vectors of all pixels inside the superpixel.

Secondly, regarding the optimization over $\theta \in \Theta$, where Θ is all the possible values of θ, Shotton et al. proposed to carefully use the histograms to give the thresholded sums necessary for the post-computation. This process might be more efficient but still involves an over-all computation in Θ. Inferred from this, we apply here another optimization process that appears to reduce the searching space of θ. The discrete set Θ contains values of $v_{[r,f]}(i)$ for all training samples, since they are all possible values of θ. It is separated into 20 bins, each bin is weighted by $\omega_i^c z_i^c$ and ω_i^c separately, and summed up to get the histogram value over these 20 bins denoted by H_1 and H_2. The difference of $|H_1 - H_2|$ is then computed to get the three bins with the most variation. Since this difference represents the amount that label changes between +1 and −1, we search in each decision stump to find a threshold which best separate +1 and −1 labels. And this threshold most probably lies in the bins which occur the strongest variation over labels. As this algorithm serves for real-time IV application, this process will largely reduce the time consuming, with a compromise of slightly decent of accuracy.

5. Experiment

5.1. Data set

The experiment was applied on our polar-image data sets which contain 21 images at 320 × 240 pixels. The Day-set includes 10 images and the Dusk-set 11 images (examples shown in Fig. 4).

Fig. 4. Acquired polarization images, where the first row was taken at dusk, while the second row was taken at daytime.

Table 1
Over-all classification accuracy using polarization.

Data	HLC	HOG-DOP	LBP-DOP	HOG-AOP	LBP-AOP
Day-set	11.34	11.27	10.74	11.27	10.97
Dusk-set	13.36	11.35	**10.21**	11.26	10.73

The Dusk-set used 6 images for training and 5 images for testing, while the Day-set used 6 images for training and 4 images for testing. These images were labeled using LableME [24]. We defined 6 classes being car, road, tree, sky, building, and grass. Pixels which do not correspond to any of these classes are referred to as void class (see points in black in column (b) of Fig. 8). After superpixel segmentation, the label of any superpixel was assigned to the label describing the majority of the pixels it contains. When training the classifier, superpixels which correspond to a void class was taken off from the training data. Up to error evaluation, testing samples with void class were also disregarded.

5.2. Comparison Day-set and Dusk-set

For each image, 200 SLIC superpixels were extracted with compactness of 20. For the classification, we applied a sampling rate of $\zeta = 0.01$ using 500 boosting rounds. Other methods as [10,13] ran till 5000 boosting rounds using sampling rate $\zeta = 0.003$. To our case, we applied only 500 boosting rounds, since the classification error converged in 300–400 boosting rounds as our data set is relatively small. We used a sampling rate $\zeta = 0.01$ to examine more feature every turn to make sure that after 500 turns, every feature has got comprehensive comparison.

The over-all error accuracy is shown in Table 1, where HOG-DOP stands for combination of HOG-DOP feature with HLC feature, and also similarly in LBP-DOP, HOG-AOP and LBP-AOP.

Using only the HLC features, Day-set occurs smaller error rate (11.34%) than Dusk-set (13.36%). Using the same algorithm, even Dusk-set got more images in training, its performance still falls behind the result in Day-set. This is probably due to the poor illumination conditions for the Dusk-set. On the performance of polarization, it is not surprise to found that by adding polarization feature, result over Day-set does not vary a lot with an improvement of 0.6%, while the Dusk-set polarization features improves the result up to 3.14%. At Day-time, lights acquired by polarization imaging system are of a reflection chaos, since they are from various objects as second or even third hand reflection. While in the Dusk-set, when the illumination goes done and complex reflection appears to be less noticeable, the acquired lights' polarization properties become more coherent and less noisy. Thus, for the Dusk-set, polarization features give good performance. This shows that polarization may potentially provide a solution for improving the scene understanding ability for intelligent devices under poor illumination. This should be further evaluated with a larger data set and faster implementation.

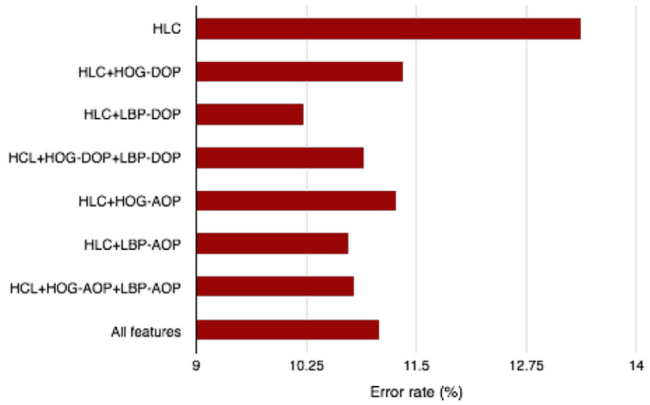

Fig. 5. The error rates of the system using different configuration of polarization features.

From Table 1, we can find that every polarization feature does improve the classification result in the Dusk-set.

5.3. Polarization features

Considering four different polarization features can be extracted, the performance of them were assessed independently. To this end, the dusk set were taken as an example.

The classification error rates with respect to different feature configurations are summarized in Fig. 5. On the *y*-axis of the figure, the configuration HLC+HOG-DOP means the combination of HLC feature with HOG on DOP image, while HLC+HOG-DOP+LBP-DOP refers to the combination of HLC feature with HOG and LBP on DOP image, and similar for the following ones. It can be observed that the LBP-DOP polarization feature always produces the best result. This confirms the result shown in Table 1, where from both of the two data sets, LBP-DOP based polarization features give better performance than other features. It should also be noted that adding more features does not lead to the improvement of classification performance. Using all the four polarization features even more brought more error than using only the HOG-DOP feature in this experiment. One reason for this may be that other features

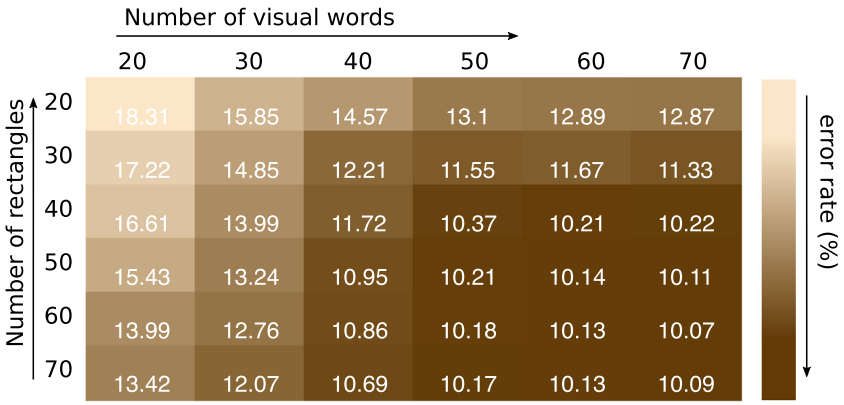

Fig. 6. The error rates with respect to different number of rectangles and different number of visual words.

Table 2
The influence of the number of visual words on the system accuracy and time consuming.

Number of visual words	Error rate (%)	10 turns time consuming	Number of turns for convergence	Over-all time consuming (min)
30	13.24	23.11	300	11
50	10.21	37.21	350	21.7
70	10.11	55.67	500	46.4
90	10.13	67.90	700	79.3

produce similar information as the HOG-DOP one, while the HOG-DOP feature contains less noise and is more robust.

Based on the above analysis, we can conclude that the best polarization-based result is found with the LBP-DOP feature. On the other hand, regarding to the time efficiency, using less feature also encourages further applications, such as embedding the system into the IV.

5.4. Parameter analysis

At the second step of the algorithm, we used 50 clustering centers (visual words) to form the codebook. Then, at the third step, 50 rectangles were randomly generated for the feature-layout-filters. In Fig. 6, the sensitiveness of the system to these two parameters was studied. It can be observed that the classification performance degrades critically if both of the two parameters are less than 40. Up to 50, the system becomes almost stable. The error rate also decreases only slightly by increasing any of this two parameters. It is noticed that by changing the number of visual words, the system is more sensitive, as the error rate decreases from 18.31% to 12.87%. While it decreases only 4.89% with respect to the number of rectangles.

Naturally, we tend to use a large number of visual words and of rectangles to ensure the segmentation accuracy. However, as the cost, too large number of visual words and rectangles may also increase the system time consuming simultaneously. As an example, the influence of the number of visual words on both system accuracy and time consuming are summarized and shown in Table 2. It can be observed that by increasing this parameter, the training time required in each turn, and also the training turn needed until the convergence, are linearly increased. As a consequence, the over-all time consuming increases exponentially with respect to the parameter.

For this reason, the numbers of rectangles and of visual words were set as 50 in our experiment, as with this value, we achieved almost the best classification performance with the least scarification on the system time consuming. It should be note that these two parameters are configured for this data set. For other data set with more training images or more classes, they should also be tuned specifically.

Another important parameter in this system is the number of the superpixels extracted from each image. Similar to the previous two parameters, the growth of this number may exponentially increase the time consuming. Therefore, the system performance as a function of the number of superpixels was also analyzed and was shown in Fig. 7. It can be found that the system accuracy increases gradually following the augmentation of this parameter, and then becomes more or less stable around 200. With a larger value of this parameter, we may further improve the segmentation

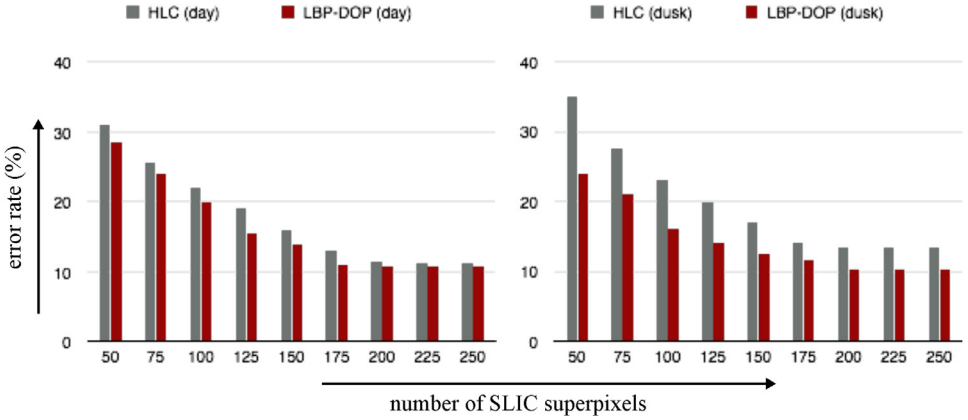

Fig. 7. The error rates of the system by using different configurations of superpixels.

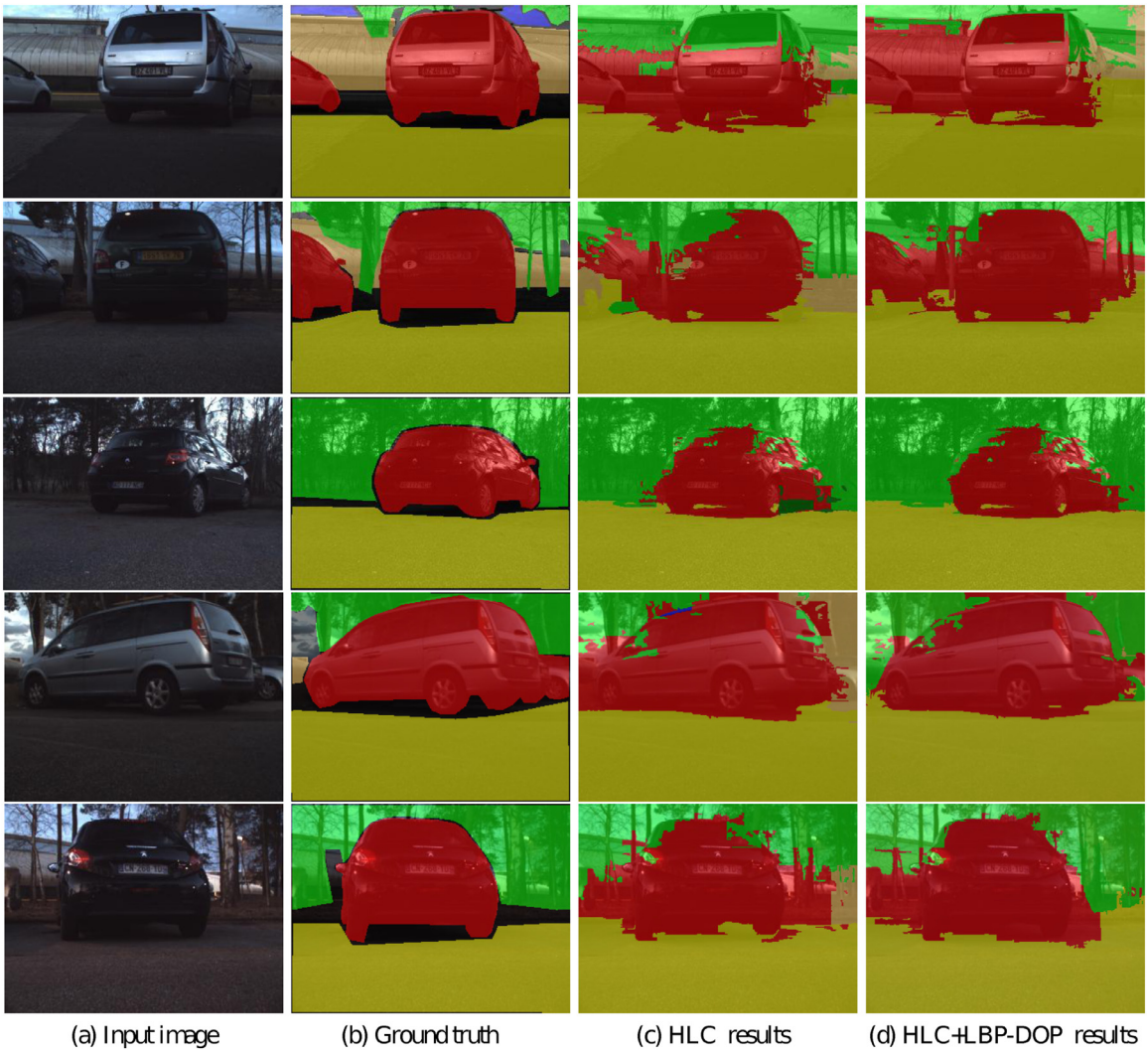

Fig. 8. Semantic segmentation result over the Dusk-set. Each row corresponds to an image scene. The column (a) shows original color images with its hand labeled ground truth on column (b). Column (c) shows the classification result using HLC and column (d) shows the HLC combined with LBP-DOP feature.

performance, while much more time consuming will be a challenge at the same time.

5.5. Visual results

The comparison between HLC feature and LBP-DOP combined feature is shown in Fig. 8. It is noted that, when applying HLC feature on the first and second scenes, the miss-classification occurred on the back window of the car. In the first scene, it might be influenced by a light reflection, while in the second scene it could be the transparency of the window. These problems were more appropriately resolved using polarization features. In the third image scene, for the HLC result, a small part of the car which has very dark intensity is miss-classified as grass; while, thanks to polarization, this problem was well addressed and better contours were obtained.

Regarding the building part in the first and second scenes, it is found that the building is merged by the car class. The reason could be that the building in the image is covered with some slightly reflecting materials, thus its polarization features would go similar with metal surface. We can also observe from the second and third scenes, over the right part of the car, that polarization features do not contribute to improve the result. As these two surfaces are of sharp angles with image plane, we supposed that polarization feature works better to the surfaces within small angles between the image plane, however, further experiment need to be done.

In fact, the semantic segmentation is a high level task in the domain of artificial intelligence. Besides the pixel classification method, it usually integrates the result of other tasks, such as the color information, the spatial position of the pixel, the edge detection, or the contextual information etc. In this paper, we propose to improve the pixel classification results through polarization, so that the experimentation was a comparison on the raw pixel classification results. It should be noted that this result can still be improved by considering other information mentioned above.

6. Conclusion

In this paper, we have proposed a method to apply polarization image on semantic segmentation. The HOG, LBP and LAB features have been extracted from polarization images, being DOP and AOP. These features have been concatenated with the color-based features as the input of the joint boosting classifier. This classifier has been used since it adapts well to combine different features, since it is principally a feature-selection based classifier. In this way, the polarization-based feature has been automatically fused with the color-based features.

In the experimentation, the comparison between the polarization-based method and the color-only method has been carried out, as well as the comparison of the results between the day-set and the dusk-set. The result using different numbers of polarization features, as well as different configuration of the system parameters are also analyzed. The experiments have shown that the polarization could improve the result of the scene understanding, especially for the dusk set. This implies that polarization might be a potential solution for intelligent devices under poor illumination.

Considering that the result of this paper is given on a small data set, it is regarded as a first attempt to apply polarization on semantic segmentation. The method should be further evaluated on a larger data set, which also needs faster implementation (such as implementation on GPU) to produce more general and efficient results.

References

[1] S. Gupta, P. Arbeláez, R. Girshick, J. Malik, Indoor scene understanding with RGB-D images: bottom-up segmentation, object detection and semantic segmentation, Int. J. Comput. Vis. 112 (2) (2015) 133–149.
[2] J. Tighe, S. Lazebnik, Superparsing, Int. J. Comput. Vis. 101 (2) (2013) 329–349.
[3] H. Zhu, F. Meng, J. Cai, S. Lu, Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation, J. Visual Commun. Image Represent. 34 (2016) 12–27.
[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.
[5] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[6] A. Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, K. Siddiqi, Turbopixels: fast superpixels using geometric flows, IEEE Trans. Pattern Anal. Mach. Intell. 31 (12) (2009) 2290–2297.
[7] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the seventh IEEE International Conference on Computer vision, 1999, 2, IEEE, 1999, pp. 1150–1157.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, 1, IEEE, 2005, pp. 886–893.
[9] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987, doi:10.1109/TPAMI.2002.1017623.
[10] J. Shotton, M. Johnson, R. Cipolla, Semantic texton forests for image categorization and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, IEEE, 2008, pp. 1–8.
[11] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1915–1929.
[12] A. Torralba, K.P. Murphy, W.T. Freeman, Sharing features: efficient boosting procedures for multiclass object detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2004, 2, IEEE, 2004, pp. II–762.
[13] A.D. Costea, S. Nedevschi, Multi-class segmentation for traffic scenarios at over 50 fps, in: Proceedings of the IEEE Intelligent Vehicles Symposium, IEEE, 2014, pp. 1390–1395.
[14] L.B. Wolff, Polarization vision: a new sensory approach to image understanding, Image Vis. Comput. 15 (2) (1997) 81–93.
[15] S. Ainouz, J. Zallat, A. De Martino, Clustering and color preview of polarization encoded images, Proceedings of the European Signal Processing Conference EUSIPCO 2006, 2006.
[16] O. Morel, C. Stolz, F. Meriaudeau, P. Gorria, Active lighting applied to three-dimensional reconstruction of specular metallic surfaces by polarization imaging, Appl. Opt. 45 (17) (2006) 4062–4068.
[17] Y. Zhao, L. Zhang, D. Zhang, Q. Pan, Object separation by polarimetric and spectral imagery fusion, Comput. Vis. Image Underst. 113 (8) (2009) 855–866.
[18] F. Qin, J. Guo, F. Lang, Superpixel segmentation for polarimetric SAR imagery using local iterative clustering, IEEE Geosci. Remote Sens. Lett. 12 (1) (2015) 13–17.
[19] fluxdata, Overview polarization camera, 2009, (http://www.fluxdata.com/-products/fd-1665-polarization-camera/).
[20] M. Born, E. Wolf, Principles of optics: electromagnetic theory of propagation, interference and diffraction of light, CUP Archive, 2000.
[21] L.B. Wolff, Surface orientation from polarization images, in: Proceedings of the IEEE Conference of the Industrial Electronics Society IECON'87, International Society for Optics and Photonics, 1988, pp. 110–121.
[22] L. Ladick'y, C. Russell, P. Kohli, P.H.S. Torr, Associative hierarchical CRFS for object class image segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 739–746.
[23] F.C. Crow, Summed-area tables for texture mapping, ACM SIGGRAPH Comput. Graph. 18 (3) (1984) 207–212.
[24] B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, Labelme: a database and web-based tool for image annotation, Int. J. Comput. Vis. 77 (1–3) (2008) 157–173.

Fan Wang received the B.S. degree in Electronic and Information Engineering from the Xidian University, Xi'an, China. She is currently pursuing the Ph.D. degree with the Laboratory LITIS, INSA de Rouen, France. Her current research interests concern the applications of the Polarization image in computer vision and intelligent vehicle.

Samia Ainouz received her Ph.D. degree in image processing from Louis Pasteur University, Strasbourg. She carried out her postdoctoral work in 3D vision at Le2i UMR 6306 CNRS Lab. Since September 2008, she has worked as an associate professor in the LITIS Lab with the Intelligent Transportation Systems Team. Her main research interests are polarization imaging, stereovision, catadioptric vision, and applications of these technics to intelligent vehicles.

Chunfeng Lian is currently pursuing the Ph.D. degree with the Laboratory LITIS University of Rouen, France. His research interests include information fusion, pattern recognition, and medical image analysis.

Abdelaziz Bensrhair graduated with an M.Sc in electrical engineering (1989) and a Ph.D. degree in computer science (1992) at the University of Rouen, France. From 1992 to 1999, he was an assistant professor in the Physic and Instrumentation Department, University of Rouen. He is currently a professor in information systems architecture department, head of Intelligent Transportation Systems Division and co-director of the Computer Science, Information Processing, and Systems, Laboratory (LITIS) of the National Institute of Applied Science Rouen (INSAR).